







This screenshot shows the Visual Studio Code editor with the file Explorer on the left displaying a project structure for 'Challenge2'. The Explorer shows folders for 'Assets', 'Scripts', 'Course Library', 'Scenes', 'Logs', 'Packages', and 'UserSettings'. The 'Scripts' folder is expanded, showing several C# files including 'DestroyOutOfBoundsX.cs', 'DetectCollisionsX.cs', 'MoveForwardX.cs', 'PlayerControllerX.cs', and 'SpawnManagerX.cs'. The main editor window displays the 'DestroyOutOfBoundsX.cs' script, which is a C# class inheriting from 'MonoBehaviour'. It contains two private float variables, 'leftLimit' (30) and 'bottomLimit' (-5), and an 'Update()' method. The 'Update()' method contains two conditional checks: one for 'transform.position.x < leftLimit' which calls 'Destroy(gameObject)', and another for 'transform.position.z < bottomLimit' which also calls 'Destroy(gameObject)'. The status bar at the bottom indicates 'Ln 14, Col 35', 'Spaces: 4', 'UTF-8 with BOM', 'LF', 'C#', 'Go Live', and 'Prettier'.

```
Assets > Challenge2 > Scripts > DestroyOutOfBoundsX.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DestroyOutOfBoundsX : MonoBehaviour
6  {
7      private float leftLimit = 30;
8      private float bottomLimit = -5;
9
10     // Update is called once per frame
11     void Update()
12     {
13         // Destroy dogs if x position less than left limit
14         if (transform.position.x < leftLimit)
15         {
16             Destroy(gameObject);
17         }
18         // Destroy balls if y position is less than bottomLimit
19         else if (transform.position.z < bottomLimit)
20         {
21             Destroy(gameObject);
22         }
23     }
24 }
25
26
```

This screenshot shows the Visual Studio Code editor with the file Explorer on the left, similar to the first image. The main editor window displays the 'SpawnManagerX.cs' script. It is a C# class with two private float variables: 'startDelay' (1.0f) and 'spawnInterval' (4.0f). It has a 'Start()' method that calls 'InvokeRepeating("SpawnRandomBall", startDelay, spawnInterval);'. There is also a 'SpawnRandomBall()' method. Inside 'SpawnRandomBall()', it generates a random ball index using 'Random.Range(0, ballPrefabs.Length)', creates a 'Vector3' for 'spawnPos' with random x and z coordinates and y=0, and then calls 'Instantiate()' with 'ballPrefabs[ballIndex]' at the 'spawnPos' with its rotation. The status bar at the bottom indicates 'Ln 32, Col 1', 'Spaces: 4', 'UTF-8 with BOM', 'LF', 'C#', 'Go Live', and 'Prettier'.

```
Assets > Challenge2 > Scripts > SpawnManagerX.cs
6
7 {
8     private float startDelay = 1.0f;
9     private float spawnInterval = 4.0f;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         InvokeRepeating("SpawnRandomBall", startDelay, spawnInterval);
15     }
16
17     // Spawn random ball at random x position at top of play area
18     void SpawnRandomBall ()
19     {
20         int ballIndex = Random.Range(0, ballPrefabs.Length);
21
22         // Generate random ball index and random spawn position
23         Vector3 spawnPos = new Vector3(Random.Range(spawnLimitXLeft, spawnLimitXRight), spawnPosY, 0);
24
25         // instantiate ball at random spawn location
26         Instantiate(ballPrefabs[ballIndex], spawnPos, ballPrefabs[ballIndex].transform.rotation);
27     }
28 }
29
30
31
32
33
34
35
36
```